

## AMENDMENTS IN THE CLAIMS

1. (Currently Amended) A method for generating a primary scrambling code, the method comprising the steps of:

generating a first m-sequence from a first m-sequence generator including first shift registers having first shift register values  $a_i$ , wherein  $i = 0$  to  $c-1$  and where  $c$  is the total number of the registers;

generating a second m-sequence from a second m-sequence generator including second shift registers having values  $b_j$ , wherein  $j = 0$  to  $c-1$ , and where  $c$  is the total number of the registers;

masking the first shift register values  $a_i$  with a first set of mask values  $K_i$ , wherein  $i = 0$  to  $c-1$  to generate a third m-sequence;

adding the first m-sequence with the second m-sequence to generate a primary scrambling code; and

adding the third m-sequence and the second m-sequence to generate a secondary scrambling code;

wherein, the masking step ~~shifts~~ is adapted to shift the first m-sequence cyclically by  $L$  chips to generate an  $L^{\text{th}}$  secondary scrambling code associated with the primary scrambling code.

2-20. (Cancelled)

21. (Currently Amended) A scrambling code generator, comprising:

a first m-sequence generator to generate a first m-sequence by using a plurality of first registers with first shift register values  $a_i$ , wherein  $i = 0$  to  $c-1$  and where  $c$  is the total number of the first registers;

a second m-sequence generator to generate a second m-sequence by using a plurality of second registers with second shift register values  $b_j$ , wherein  $j = 0$  to  $c-1$  and where  $c$  is the total number of second registers;

a masking section to mask the first shift register values  $a_i$  with a first set of mask values  $K_i$  to generate a third m-sequence, wherein  $i = 0$  to  $c-1$  to generate a third m-sequence;

a first adder to add the first m-sequence and the second m-sequence to generate a primary scrambling code; and

a second adder to add the third m-sequence and the second m-sequence to generate a secondary scrambling code,

wherein the masking section ~~shifts~~ is adapted to shift the first m-sequence cyclically by  $L$  chips to generate an  $L^{\text{th}}$  secondary scrambling code associated with the primary scrambling code.

22-30. (Cancelled)

31. (Previously Presented) The method of claim 1, wherein the primary scrambling code is one of a plurality primary scrambling codes and a  $K^{\text{th}}$  primary scrambling code is a  $((K-1)*M+K)^{\text{th}}$  gold code, where  $M$  is a total number of secondary scrambling codes per primary scrambling code and  $1 \leq K \leq 512$ .

32. (Previously Presented) The method of claim 1, wherein the secondary scrambling codes associated with a  $K^{\text{th}}$  primary scrambling code are from  $((K-1)*M+K+1)^{\text{th}}$  to  $(K*M+K)^{\text{th}}$  gold codes, where  $M$  is a total number of secondary scrambling codes per primary scrambling code and  $1 \leq K \leq 512$ .

33. (Previously Presented) The method of claim 1, wherein  $1 \leq L \leq M$ , where  $M$  is a total number of secondary scrambling codes per primary scrambling code.

34. (Previously Presented) The method of claim 1, wherein the masking step is expressed by  $\sum (k_i \times a_i)$ .

35. (Previously Presented) The method of claim 1, further comprising:

masking the first shift register values  $a_i$  with a second set of mask values  $K_j$  to generate a fourth m-sequence, wherein  $j = 0$  to  $c-1$ ; and

adding the fourth m-sequence and the second m-sequence to generate an  $N^{\text{th}}$  secondary scrambling code associated with the primary scrambling code;

wherein, the masking step shifts the first m-sequence cyclically by N chips to generate an N<sup>th</sup> secondary scrambling code.

36. (Previously Presented) The method of claim 35, wherein  $1 \leq N \leq M$ , where M is a total number of secondary scrambling codes per primary scrambling code.

37. (Previously Presented) The method of claim 1, further comprising the step of delaying at least one of the primary scrambling code and secondary scrambling code to produce a Q-channel component, wherein the primary scrambling code and secondary scrambling code are I-channel components.

38. (Previously Presented) The scrambling code generator of claim 21, wherein the primary scrambling code is one of a plurality of primary scrambling codes and a K<sup>th</sup> primary scrambling code is a  $((K-1)*M+K)^{th}$  gold code, where M is a total number of secondary scrambling codes per primary scrambling code and  $1 \leq K \leq 512$ .

39. (Previously Presented) The scrambling code generator of claim 38, wherein the secondary scrambling codes associated with the K<sup>th</sup> primary scrambling code are  $((K-1)*M+K+1)^{th}$  to  $(K*M+K)^{th}$  gold codes.

40. (Currently Amended) The scrambling code generator of claim 21, further comprising:

a second masking section to mask the first shift register values  $a_i$ , with a second set of mask values  $K_j$ , wherein  $j = 0$  to  $c-1$ , to generate a fourth m-sequence; and

a third adder to add the fourth m-sequence and the second m-sequence to generate an N-th secondary scrambling code associated with the primary scrambling code,

wherein the second masking section ~~shifts is adapted to shift~~ the first m-sequence cyclically by N chips to generate the N<sup>th</sup> secondary scrambling code.

41. (Currently Amended) The scrambling code generator of claim 21, wherein the masking section ~~shifts is adapted to shift~~ the first m-sequence cyclically by masking the first shift register values  $a_i$  in accordance with  $\sum (K_i \times a_i)$ .

42. (Currently Amended) The scrambling code generator of claim 21, wherein the first m-sequence generator cyclically shifts ~~is adapted to cyclically shift~~ the first shift register values and the second m-sequence generator cyclically shifts ~~is adapted to cyclically shift~~ the second shift register values.

43. (Currently Amended) The scrambling code generator of claim 21, wherein the first m-sequence generator adds ~~is further adapted to add~~ predetermined shift register values of the first shift registers based on a first generating polynomial of the first m-sequence, right shifts ~~shifting~~ the first shift register values  $a_i$  of the first shift registers, and replaces ~~replacing~~ the first register value  $a_{c-1}$  with the result of the addition of the predetermined register values.

44. (Currently Amended) The scrambling code generator of claim 21, wherein the first m-sequence generator adds ~~is adapted to add~~ a first shift register value  $a_0$  with a first shift register  $a_7$  to form a next first shift register  $a_{c-1}$ .

45. (Currently Amended) The scrambling code generator of claim 21, wherein the second m-sequence generator adds ~~is further adapted to add~~ predetermined shift register values of the second shift registers based on a second generating polynomial of the second m-sequence, right shifts ~~shifting~~ the second shift register values  $b_j$  of the second shift registers, and replaces ~~replacing~~ the second register value  $b_{c-1}$  with the result of the addition of the predetermined register values.

46. (Currently Amended) The scrambling code generator of claim 21, wherein the second m-sequence generator adds ~~is adapted to add~~ a second shift register value  $b_0$  with a second shift register value  $b_5$ ,  $b_7$ , and a second shift register value  $b_{10}$  to form a next second shift register value  $b_{c-1}$ .

47. (Previously Presented) The apparatus of claim 21, further comprising a means for delaying at least one of the primary scrambling code and the secondary scrambling code to

produce Q-channel component, wherein the primary scrambling code and the secondary scrambling code are I-channel components.

48. (Currently Amended) A method for generating scrambling codes in a mobile communication system, comprising:

generating a primary scrambling code for a first base station by adding a first m-sequence and a second m-sequence;

generating a secondary scrambling code for the first base station, the generated secondary scrambling code belonging to a scrambling codes group having  $((K-1)*M+K)^{th}$  through  $(K*M+K)^{th}$  scrambling codes assigned to the first base station and having  $(K*M+K+1)^{th}$  through  $((K+1)*M+K+1)^{th}$  scrambling codes assigned to a second base station, wherein M is a total number of secondary scrambling codes per primary scrambling code and K is a natural number, the  $((K-1)*M+K)^{th}$  and  $(K*M+K+1)^{th}$  scrambling codes being the primary scrambling code of the first base station and a primary scrambling code of the second base station, respectively, and the  $((K-1)*M+K+1)^{th}$  through  $((K+1)*M+K+1)^{th}$  scrambling codes are generated by shifting the first m-sequence by  $((K-1)*M+K)$  through  $((K+1)*M+K)$  chips, respectively; and

adding the shifted first m-sequence to the second m-sequence, wherein the  $((K-1)*M+K+1)^{th}$  through  $((K+1)*M+K+1)^{th}$  scrambling codes that are generated by shifting the first m-sequence by  $((K-1)*M+K)$  through  $((K+1)*M+K)$  chips, respectively, are used to separate the  $((K-1)*M+K)^{th}$  through  $(K*M+K)^{th}$  scrambling codes assigned to the first base station from the  $(K*M+K+1)^{th}$  through  $((K+1)*M+K+1)^{th}$  scrambling codes assigned to the second base station.

49. (Previously Presented) The method of claim 48, wherein the step of generating the primary scrambling code of the first base station comprises:

generating the first m-sequence with a first m-sequence generator, the first m-sequence generator including a first group of shift registers, wherein each of the shift registers is configured to store a bit of data; and

generating the second m-sequence with a second m-sequence generator, the first m-sequence generator including a second group of shift registers, wherein each of the shift registers is configured to store a bit of data.

50. (Previously Presented) The method of claim 48, wherein the  $((K-1)*M+K+1)^{\text{th}}$  through  $((K+1)*M+K+1)^{\text{th}}$  scrambling codes generated by shifting of the first m-sequence by  $((K-1)*M+K)$  through  $((K+1)*M+K)$  chips, respectively, are used to save a storage space for storing scrambling codes for a plurality of base stations.

51. (Currently Amended) A method for managing scrambling code assignment, comprising:

~~assigning, to managing~~ a first base station, ~~which is assigned~~ a primary scrambling code and secondary scrambling codes of a scrambling codes group having  $((K-1)*M+K)^{\text{th}}$  through  $(K*M+K)^{\text{th}}$  scrambling codes assigned to the first base station and having  $(K*M+K+1)^{\text{th}}$  through  $((K+1)*M+K+1)^{\text{th}}$  scrambling codes assigned to a second base station, wherein M is a total number of secondary scrambling codes per primary scrambling code and K is a natural number, the  $((K-1)*M+K)^{\text{th}}$  and  $(K*M+K+1)^{\text{th}}$  scrambling codes being the primary scrambling code of the first base station and a primary scrambling code of the second base station, respectively, the  $((K-1)*M+K+1)^{\text{th}}$  through  $((K+1)*M+K+1)^{\text{th}}$  scrambling codes are generated by shifting a first m-sequence by  $((K-1)*M+K)$  through  $((K+1)*M+K)$  chips, respectively, and adding the shifted first m-sequence to a second m-sequence;

~~managing the second base station;~~ and

generating the primary scrambling sequence and a secondary scrambling code of at least one of the first base station and a ~~the~~ second base station, wherein the  $((K-1)*M+K+1)^{\text{th}}$  through  $((K+1)*M+K+1)^{\text{th}}$  scrambling codes that are generated by shifting the first m-sequence by  $((K-1)*M+K)$  through  $((K+1)*M+K)$  chips, respectively, are used to separate the  $((K-1)*M+K)^{\text{th}}$  through  $(K*M+K)^{\text{th}}$  scrambling codes assigned to the first base station from the  $(K*M+K+1)^{\text{th}}$  through  $((K+1)*M+K+1)^{\text{th}}$  scrambling codes assigned to the second base station.

52. (Previously Presented) The method of claim 51, wherein the step of generating the primary scrambling code and the second scrambling code of the at least one of the first base station and the second base station comprises:

generating the first m-sequence with a first m-sequence generator, the first m-sequence generator including a first group of shift registers, wherein each of the shift registers stores a bit of data; and

generating the second m-sequence with a second m-sequence generator, the second m-sequence generator including a second group of shift registers, wherein each of the shift registers stores a bit of data.

53. (Previously Presented) The method of claim 51, wherein the  $((K-1)*M+K+1)^{\text{th}}$  through  $((K+1)*M+K+1)^{\text{th}}$  scrambling codes that are generated by shifting of the first m-sequence by  $((K-1)*M+K)$  through  $((K+1)*M+K)$  chips, respectively, are used to save a storage space for storing scrambling codes for a plurality of base stations.